IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	:	10/710,477 Confirmation No.: 4476
Applicant:	:	James E. Aston et al.
Filed:	:	July 14, 2004
Title:	:	Method and System to Protect a File System from
	:	Viral Infections
TC/A.U.	:	2168
Examiner:	:	Dwivedi, Mahesh H..
Docket No.	:	014682.000010
Customer No.	:	44,870

Mail Stop: AF
Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF IN COMPLIANCE WITH 37 CFR 41.37

In response to the Notice of Panel Decision from Pre-Appeal Brief Review dated as mailed December 15, 2008 this appeal brief is being submitted. The Notice of Appeal was acknowledged as being received on Feb. 15, 2008.

I. Real Party in Interest

The real party in interest is International Business Machines (IBM) Corporation, assignee of record.

II. Related Appeals and Interferences

There are no other appeals or interferences, known to the Appellants, or Appellants' legal representatives, which will directly affect or be directly affected by or have a bearing on the Board's decision in this pending appeal.

III. Status of Claims

Claims 1-5 and 7-20 stand rejected. Claims 6 and 21-44 have been cancelled.

IV. Status of Amendments

An amendment was filed on February 15, 2008 after the final Office Action dated as mailed November 16, 2007. The amendment cancelled Claims 21-44. The amendment was entered according to the Advisory Action dated as mailed May 12, 2008. Despite the Advisory Action, the Notice of Panel Decision from Pre-Appeal Brief Review indicated that Claims 21-44 have been rejected. For purposes of this appeal, Applicants' legal representatives assume Claims 21-44 have been properly cancelled and are not being appealed. All previous papers filed by Applicants have been entered.

V. Summary of Claimed Subject Matter

The present invention is related to protecting a file system from a viral infection. A program running on a computer is flagged as being suspect for possibly containing a virus in response to the program performing at least one of a set of predetermined file system operations. The file system operations include: opening a local file on a local file system of the computer to perform a read operation and opening a shared file on a shared or network file system to perform a write or append operation with the local file; the program reading or opening itself and the program attempting to write or append any content to the shared file on the shared or network file system or to write or append any content to the local file on the local file system; the program attempting to write or append the local file to the shared or network file system and preserve a filename of the local file in the shared or network file system; and the program attempting to write or append a remote file to the local file system. A filename and a location where the local or shared file is copied or written are stored in response to the local or shared file being copied or written by the program.

Claims 1 and 9 are independent claims and stand rejected in the present application. Claims 2-5 and 7-8 are dependent claims depending directly from Independent Claim 1 and stand rejected in the present application. Claims 10-20 are dependent claims depending either directly of indirectly from Independent Claim 9 and stand rejected in the present application.

Independent Claim 1 is a method claim. Claim 1 recites, "A method to protect a file system from a viral infection, comprising: flagging a program on a computer as being suspect for

possibly containing a virus in response to at least one of: opening a local file on a local file system of the computer to perform a read operation and opening a shared file on a shared or network file system to perform a write or append operation with the local file". This feature of Claim 1 is described in the specification in paragraph [0014] and in Figure 1C blocks 126 and 130 and in Figure 1F block 134. Claim 1 also recites, "the program reading or opening itself and the program attempting to write or append any content to the shared file on the shared or network file system or to write or append any content to the local file on the local file system". This feature of Claim 1 is described in the specification in paragraph [0017] and in Figure 1D blocks 128 and 158. Claim 1 additionally recites, "the program attempting to write or append the local file to the shared or network file system and preserve a filename of the local file in the shared or network file system; and the program attempting to write or append a remote file to the local file system". These features of Claim 1 are described in the specification in paragraph [0018] and in Figure 1E blocks 156 and 164. Claim 1 further recites, "storing a filename and a location where the local or shared file is copied or written in response to the local or shared file being copied or written by the program". This feature of Claim 1 is described in paragraph [0016] and in Figure 1G block 154.

Independent Claim 9 is a method claim. Claim 9 recites, "A method to protect a file system from a viral infection, comprising: allowing a security level to be set". This feature of Claim 9 is described in the specification in paragraph [0011] and in Figure 1A block 102. Claim 9 also recites, "monitoring predetermined file system operations associated with a program". This feature of Claim 9 is described in the specification in paragraph [0013] and in Figure 1A block 116. Claim 9 further recites, "logging any predetermined file system operations associated with the program including recording a filename and a location where a file is written in response to the file being written". This feature of Claim 9 is described in the specification in paragraph [0012] and in Figure 1A block 110.

Dependent Claim 15 depends directly from dependent Claim 14 which depends directly from independent Claim 9. Claim 15 recites the same features to Claim 1 described above.

VI. Grounds of Rejection to be Reviewed on Appeal

1. Whether Claims 1-2, 5, 7-9 and 12-19 are unpatentable under 35 U.S.C. §102(b) as being anticipated by *Halperin* et al. (U.S. Pub. No. 2002/0194490; hereinafter *Halperin*).
2. Whether Claims 3-4, 10-11, and 20 are unpatentable under 35 U.S.C. §103(a) over *Halperin* in view of *Satterlee* et al. (U.S. Patent Pub. No. 2004/0025015; hereinafter *Satterlee*).

VII. Arguments

Rejection under 35 U.S.C. §102(a) as being unpatentable as being anticipated by Halperin

Claims 1-2, 5, and 7-8

For the reasons discussed herein, Applicant respectfully submits that *Halperin* fails to teach the essential elements needed for a prima facie rejection under 35 U.S.C. § 102. Claim 1 recites:

> "flagging a program on a computer as being suspect for possibly containing a virus in response to at least one of:
>
> opening a local file on a local file system of the computer to perform a read operation and opening a shared file on a shared or network file system to perform a write or amend operation with the local file;
>
> the program reading or opening itself and the program attempting to write or append any content to the shared file on the shared or network file system or to write or append any content to the local file on the local file system;
>
> the program attempting to write or append the local file to the shared or network file system and preserve a file name of the local file in the shared or network file system; and
>
> the program attempting to write or append a remote file to the local file system; . . ."

The Final Office Action dated as mailed November 16, 2007, cited the Abstract, Paragraphs 73-77, and 88-108 of *Halperin* in rejecting these features of Claim 1. The Abstract of *Halperin* recites:

> "A method for malicious software detection including grouping a plurality of computing devices in a network into at least two groups, measuring a normal operation value of at least one operating parameter of any of the groups and detecting a change in value to indicate possible malicious software behavior within the network."

Accordingly, from the abstract of *Halperin*, *Halperin* teaches detection of malicious software in a plurality of computing devices within a network and not within an individual computer as provided by the present invention. *Halperin* does not teach or suggest flagging a program on a computer as being suspect for possibly containing a virus, nor does *Halperin* teach or suggest the specific conditions associated with the computer for flagging the program on the computer as provided by the embodiment of the present invention as recited in Claim 1.

Paragraphs 73-77 of *Halperin*, also cited in the Final Office Action in rejecting Claim 1, recite:

> "[0073] Reference is now made to FIG. 2, which is a simplified flow chart illustration of an exemplary method of operation of the system of FIG. 1, useful in understand the present invention. In the method of FIG. 2, computer 100 becomes infected by a computer virus such as by receiving the virus by another computer via network 102 or via the introduction of infected data storage media such as a diskette or a compact disc into computer 100. As the virus attempts to propagate it selects one or more valid and decoy addresses from address book 102 in folders 104, automatically generates messages that incorporate the virus, typically as an attachment, and forwards the messages to server 108. Server 108 scans messages received from computer 100. Should server 108 detect a message addressed to a decoy address, server 108 may initiate one or more virus containment actions such as, but not limited to:
>
> [0074] Suspending any or all messages sent by computer 100, thereby preventing messages sent by computer 100 from being forwarded to recipients.
>
> [0075] Forwarding messages that are addressed to a decoy address to a third party for analysis, such as a company or other body that produces anti-virus software.
>
> [0076] Notifying a user at computer 100 of the suspicious message activity.
>
> [0077] Notifying a system administrator that a virus may have been detected."
>
> (emphasis added)

Thus, *Halperin* teaches sending messages across a network from a computer 100 to a server 108, scanning the messages received from computer 100 on the server 108, and if the

server 108 detects a message addressed to a decoy address, server 108 may initiate one or more virus containment actions such as those recited above from *Halperin*. *Halperin* teaches suspending messages being sent by an infected computer but *Halperin* does not teach or suggest flagging a program on the computer based on specific local file system operations as provided by the embodiment of the present invention as recited in Claim 1 of the present application. Applicant further respectfully submits that there is no teaching or suggestion in *Halperin* of the specific local file system operations being performed on the computer for determining whether to flag a program on the computer as being suspect for possibly containing a virus. *Halperin* merely teaches initiating virus scanning and virus containment actions associated with email messages at the server and network level as opposed to file system operations at the computer level or client level as required by the embodiment of the present invention as recited in Claim 1.

The Final Office Action also cited paragraphs [0088]-[0108] in rejecting the specific file system operations as recited above in Claim 1. Paragraphs [0088]-[0096] describe specific target behavior profiles and provides a list of examples. Applicant respectfully submits that paragraphs [0088]-[0096] of *Halperin* also do not teach or suggest the specific local file system operations being performed on the computer for determining whether to flag a program on the computer as being suspect for possibly containing a virus as recited in Claim 1.

Paragraph [0097] of *Halperin* beginning at line 6 and continuing through paragraph [0107] which was also cited in rejecting Claim 1, recites:

> "After collecting information regarding target behavior detected at two or more of computers 500, server 502 may then correlate the presence of target behavior detected at two or more of computers 500 in order to determine whether the correlated target behavior corresponds to a predefined suspicious behavior pattern of target behavior as an indication that a computer virus may have infected those computers. Any known behavior correlation techniques may be used, such as identifying the same activity in different computers at about the same time, or by identifying repeating patterns of data within the memories of two or more computers. Examples of expressions of such suspicious behavior patterns include:
>
> [0098] A certain percentage of the computers in the network sending more than 10 messages per minute in the last 5 minutes;
>
> [0099] A certain percentage of the computers in the network sending messages not initiated via the message GUI in the last 1 minute;
>
> [0100] A certain percentage of the computers in the network deleting more than 10 files in the last 1 minute;

[0101] A certain percentage of computers in the network deleting a file by the same name within the last 1 hour.

[0102] A certain percentage of the computers in the network deleting a file with the same name in the last 1 minute;

[0103] A certain percentage of the computers in the network to which changes to the Microsoft Windows.RTM. Registry occurred in the last 1 minute;

[0104] A certain percentage of the computers in the network sending the same file attachment via a message in the last 15 minutes;

[0105] A certain percentage of the computers in the network sending file attachments via one or more messages in the last hour where each of the files includes the same string of bits;

[0106] A certain percentage of the computers in the network having an unusual level of correlation of data between files sent as attachments. For example, since viruses known as "polymorphic viruses" may change their name as they move from one computer to another, one way to identify such viruses is to identify attachments that have the same or similar data, whether or not they have the same name.

[0107] Upon detecting a suspicious behavior pattern server 502 may initiate one or more virus containment actions such as is described hereinabove with reference to FIG. 2."

Accordingly, *Halperin* teaches collecting information regarding target behavior detected at two or more computers 500 by the server 502. Then the server 502 correlates the presence of the target behavior to determine whether the correlated target behavior corresponds to a pre-defined suspicious behavior as an indication that a computer virus may have infected those computers. Applicant respectfully submits that there is no teaching or suggestion in *Halperin* of flagging a program on an individual computer as being suspect for possibly containing a virus. Additionally, as indicated by the recitation from *Halperin* above, paragraphs [0098] – [0107] provide examples of expressions of suspicious behavior patterns. Applicant respectfully submits that none of the suspicious behavior patterns taught by *Halperin* teach or suggest the specific conditions or file system operations recited in Claim 1 for flagging the program on the individual computer.

Additionally, Claim 1 recites:

"storing a file name and a location where the local or shared file is copied or written in response to the local or shared file being copied or written by the program."

The Office Action cited paragraph [0108] of *Halperin* for rejecting this feature of Claim 1. Paragraph [0108] of *Halperin* recites:

> "[0108] In the systems and methods described hereinabove with reference to FIGS. 1, 2, 3, 4, 5, and 6, the server may include a buffer or other mechanism whereby messages received from the computer are held, typically for a predefined delay period, prior to forwarding the messages to their intended recipients. In this way, should a computer virus send one or more infected messages to valid, non-decoy addresses before sending an infected message to a decoy address, the infected messages to valid, non-decoy addresses that are still held at the server may be "quarantined" at the server and thus prevented, together with the infected message to a decoy address, from reaching their intended destinations...."

Accordingly, Applicant respectfully submits that *Halperin* teaches away from the present invention in that *Halperin* quarantines the messages and does not allow them to reach their intended destinations. In contrast, the present invention, as recited in the embodiment of Claim 1, permits the file to be copied or written by the program at its intended location and then stores the file name and location where the local or shared file is copied or written. In further distinction, *Halperin* teaches storing an entire message, not only a name or identification as in the present invention. The message in *Halperin* is not allowed to be sent to its intended destination. Accordingly, there is no need for *Halperin* to store a file name and a location where the file is copied or written as in the present invention.

For all of the reasons discussed above, Applicant respectfully submits that Claim 1 is patentably distinguishable over *Halperin*, and reconsideration and withdrawal of the 35 U.S.C. § 102 Rejection of Claim 1 is respectfully requested.

Claims 2, 5, 7 and 8 depend directly from independent Claim 1. Because of this dependency, Claims 2, 5, 7 and 8 contain all of the features of independent Claim 1. Therefore, these claims are also respectfully submitted to be patentably distinguishable over *Halperin*, and reconsideration and withdrawal of the § 102 rejection of these claims is respectfully solicited.

Claims 9, 12-14, and 16-19

With respect to the rejection of independent Claim 9 under 35 U.S.C. § 102(b) as being anticipated by *Halperin*, Claim 9 recites:

> "logging any predetermined file system operations associated with the program including recording a file name and location where a file is written in response to the file being written."

The Office Action cited paragraph [0108] of Halperin in rejecting this feature of Claim 9. As previously discussed, *Halperin* teaches in paragraph [0108] quarantining at a server messages believed to contain a virus. In contrast, the present invention as recited in Claim 9 permits the file to be written and then records the file name and the location where the file is written. As discussed above, *Halperin* does not teach or suggest this feature of the present invention. Therefore, Claim 9 is respectfully submitted to be patentably distinguishable over *Halperin*, and reconsideration and withdrawal of the 35 U.S.C. § 102 rejection of Claim 9 is respectfully requested.

With respect to the rejection of Claims 12-14 and 16-19 under 35 U.S.C. § 102(b) as being anticipated by *Halperin,* these claims recite additional features which further patentably distinguish over *Halperin.* Additionally, Claims 12-14 and 16-19 depend either directly or indirectly from Independent Claim 9, and by virtue of that dependency, contain all of the features of Independent Claim 9. Therefore, Claims 12-19 are also submitted to be patentably distinguishable over *Halperin*, and reconsideration and withdrawal of the Section 102 rejection of Claims 12-14 and 16-19 is respectfully solicited.

Claim 15

Claim 15 recites flagging a program in response to different file system operations which are substantially the same as those recited in Independent Claim 1. As discussed with respect to Independent Claim 1, *Halperin* does not teach or suggest the specific file system operations recited in Dependent Claim 15. Additionally, Claim 15 depends indirectly from Independent

Claim 9. Because of this dependency, Claim 15 includes all of the features of Claim 9 and intervening Dependent Claim 14. For all of these reasons, Claim 15 is respectfully submitted to be patentably distinct over *Halperin*, and reconsideration and withdrawal of the Section 102 rejection of Claim 15 is respectfully solicited.

Rejection under 35 U.S.C. §103(a) as being unpatentable over Halperin in view of Satterlee

Claims 3-4, 10-11, and 20

Claims 3 and 4 depend directly from Independent Claim 1 and Claims 10-11 and 20 depend either directly or indirectly from the Independent Claim 9. Because of these dependencies, these dependent claims include all of the features of the referenced independent claims. Applicant respectfully submits that *Satterlee* does not teach or suggest the features of Independent Claims 1 and 9 as previously discussed which also distinguish Claims 1 and 9 over *Halperin*. Therefore, Claims 3-4, 10-12 and 20 are respectfully submitted to be patentably distinguishable over *Halperin* and *Satterlee*, and reconsideration and withdrawal of the 35 U.S.C. § 103 rejection of these claims is respectfully solicited.

Conclusion

For the reasons discussed above, Applicant respectfully submits that the rejections standing in this application are improper. The Examiner has failed to establish a *prima fascia* case of anticipation under 35 U.S.C. §102 (b) with respect to Claims 1-2, 5, 7-9, and 12-19 and a *prima fascia* case of obviousness under 35 U.S.C. §103(a) with respect to Claims 3-4, 10-11 and 20 over the cited documents. Therefore, Applicant respectfully submits that Claims 1-5 and 7-20 are in condition for allowance. Reversal of the rejection of Claims 1-5 and 7-20 is respectfully requested.

Respectfully submitted,

Paul F. McMahan
(Applicant)

Date: January 13, 2009

By: Charles L. Moore

Charles L. Moore
Registration No. 33,742
Moore & Van Allen PLLC
P.O. Box 13706
Research Triangle Park, N.C. 27709
Telephone: (919) 286-8000
Facsimile: (919) 286-8199

VIII. Claims Appendix

1. (Previously Amended) A method to protect a file system from a viral infection, comprising:

flagging a program on a computer as being suspect for possibly containing a virus in response to at least one of:

opening a local file on a local file system of the computer to perform a read operation and opening a shared file on a shared or network file system to perform a write or append operation with the local file;

the program reading or opening itself and the program attempting to write or append any content to the shared file on the shared or network file system or to write or append any content to the local file on the local file system;

the program attempting to write or append the local file to the shared or network file system and preserve a filename of the local file in the shared or network file system; and

the program attempting to write or append a remote file to the local file system;

storing a filename and a location where the local or shared file is copied or written in response to the local or shared file being copied or written by the program.

2. (Previously Presented) The method of claim 1, further comprising inhibiting a write or append operation associated with the program in response to flagging the program.

3. (Original) The method of claim 1, further comprising monitoring all file operations associated with the program in response to the program not being in a safe list.

4. (Original) The method of claim 1, further comprising permitting selected read and write operations in response to a predefined rules table.

5. (Original) The method of claim 1, further comprising sending an alert in response to flagging the program.

6. (Canceled)

7. (Original) The method of claim 1, further comprising sending an alert to a network monitoring system in response to flagging the program.

8. (Original) The method of claim 1, further comprising logging any file system operations including recording a filename and a location where the local or shared file is written.

9. (Previously Amended) A method to protect a file system from a viral infection, comprising:

allowing a security level to be set;

monitoring predetermined file system operations associated with a program; and

logging any predetermined file system operations associated with the program including recording a filename and a location where a file is written in response to the file being written.

10. (Original) The method of claim 9, further comprising selecting the program for monitoring in response to the program not being on a safe list.

11. (Original) The method of claim 10, further comprising logging any file system operations associated with any programs on the safe list.

12. (Original) The method of claim 9, further comprising receiving a notification that the program intends to perform one of the predetermined file system operations.

13. (Previously Presented) The method of claim 9, further comprising following a predefined procedure in response to the level of security set.

14. (Original) The method of claim 9, further comprising flagging the program in response to the program attempting to perform one of the predetermined file system operations.

15. (Original) The method of claim 14, further comprising flagging the program in response to at least one of:

the program opening a local file on a local file system to perform a read operation and opening a shared file on a shared or network file system to perform a write or append operation with the local file;

the program reading or opening itself and the program attempting to write or append any content to the shared file on the shared or network file system or to write or append any content to the local file on the local file system;

the program attempting to write or append the local file to the shared or network file system and preserve a filename of the local file in the shared or network file system; and

the program attempting to write or append a remote file to the local file system.

16. (Original) The method of claim 14 , further comprising inhibiting any predetermined file system operations associated with the program in response to the program being flagged.

17. (Original) The method of claim 9, further comprising sending an alert in response to the program attempting to perform any predetermined file system operations.

18. (Original) The method of claim 17, further comprising sending the alert to a network monitoring system.

19. (Original) The method of claim 9, further comprising presenting an alert to a user for approval before the predetermined file system operation is performed by the program.

20. (Previously Amended) The method of claim 9, further comprising requiring approval before performing any predetermined file system operations associated with the program in response to the program not being on a safe list.

IX. Evidence Appendix

None.

X. Related Proceedings Appendix

None.